cs2ss manuscript analysis

```
suppressPackageStartupMessages(require(viridis))
suppressPackageStartupMessages(library(caret))
knitr::opts_knit$set(root.dir = normalizePath("../.."))
```

source user functions

```
source("scripts/analysis/functions.R")
```

Figure 1: Neural network illustration

```
draw_neural_network()
```

```
## Loading required package: scales
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:viridis':
##
##     viridis_pal
```

```
## Warning in plot.nnet(mod, node.labs = FALSE, circle.col = "blue", pch =
## 21, : Bias layer not applicable for rsnns object
```

```
## Loading required package: reshape
```

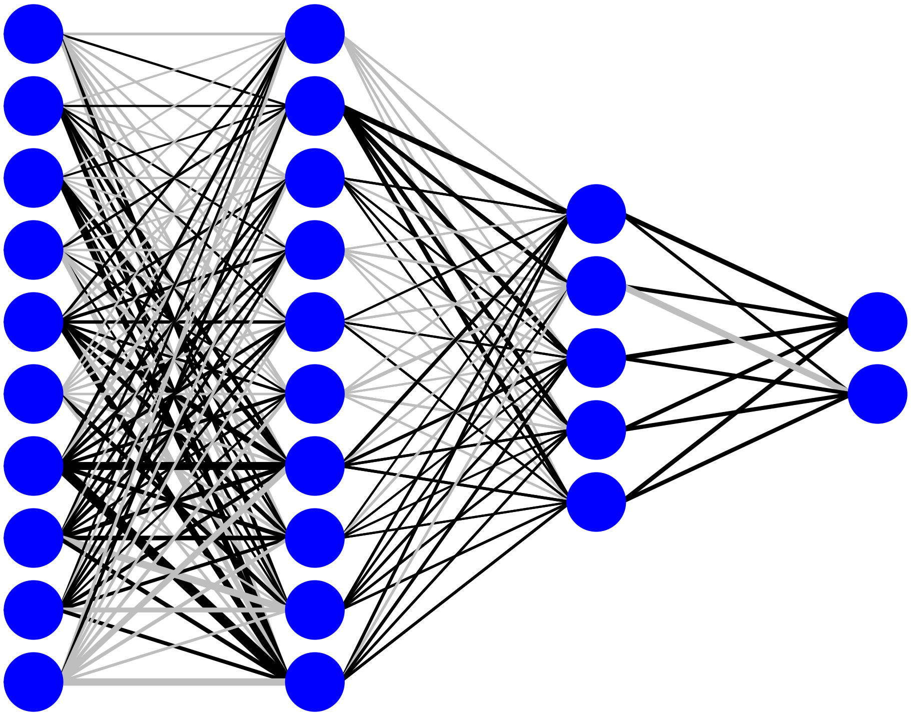


Figure 2: CS2BPS assessment

load cs2bps derived classifier assessment

```
data <- load_model_accuracy()
print_model_summary(data, metrics = c("sen", "spec", "overall"))
```

```
## metric:     sen min: 0.77 max: 1.00 mean: 0.95 median: 0.97
## metric:    spec min: 0.00 max: 1.00 mean: 0.72 median: 0.75
## metric: overall min: 0.63 max: 1.00 mean: 0.88 median: 0.88
```

plot cs2bps TPR

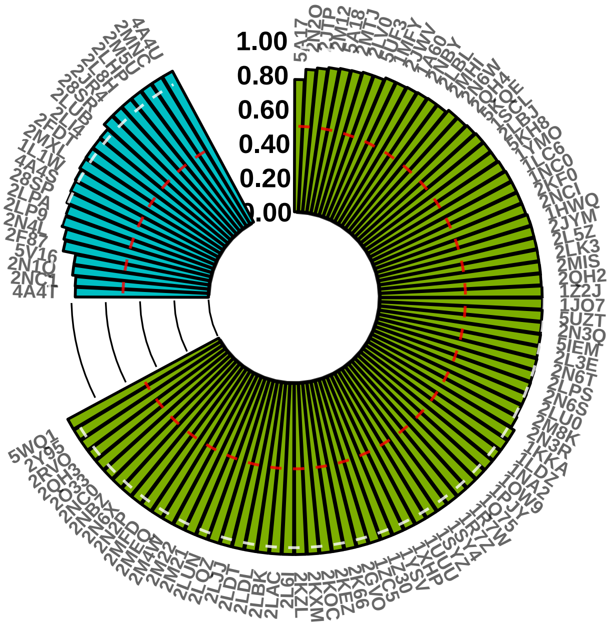
```
plot_color_bar(data, "sen")
```

```
## Source: local data frame [2 x 5]
## Groups: <by row>
##
## # A tibble: 2 x 5
##   group  start   end title  mean
##   <fct>  <dbl> <dbl> <dbl> <dbl>
## 1 both       1    86  43.5 0.960
## 2 proton    97   118 108.  0.929
```

```
## Warning: Removed 20 rows containing missing values (position_stack).
```

```
## Warning: Removed 20 rows containing missing values (position_stack).
```

```
## Warning: Removed 20 rows containing missing values (geom_text).
```



plot cs2bps TNR

```
plot_color_bar(data, "spec")
```

```
## Source: local data frame [2 x 5]
## Groups: <by row>
##
## # A tibble: 2 x 5
##   group  start   end title  mean
##   <fct>  <dbl> <dbl> <dbl> <dbl>
## 1 both       1    86  43.5 0.732
## 2 proton    97   118 108.  0.673
```

```
## Warning: Removed 20 rows containing missing values (position_stack).

## Warning: Removed 20 rows containing missing values (position_stack).
```

```
## Warning: Removed 20 rows containing missing values (geom_text).
```

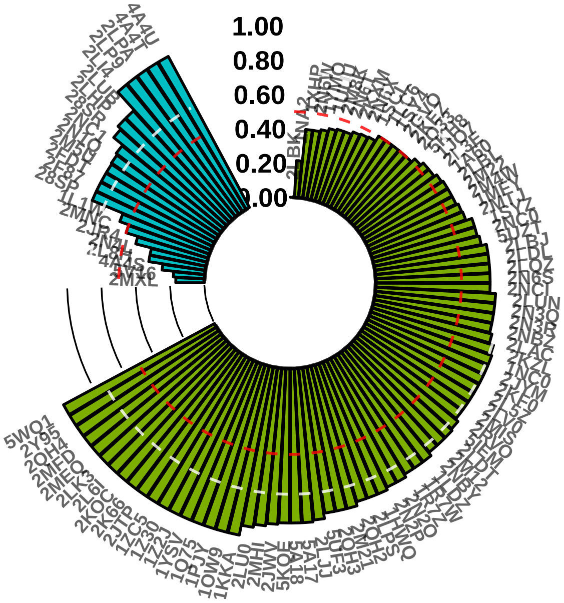


Figure 3: CS-Folding assessment

load TPR and PPV of heuristically selected secondary structures

```
# did not calculate consistency score
data <- load_model_accuracy(file = "data/ss_scorer/heuristic_scorer_summary.txt", colnames = c("id", "s
data$sens <- as.numeric(gsub("*%","",as.character(data$sens)))/100
data$PPV <- as.numeric(gsub("*%","",as.character(data$PPV)))/100
print_model_summary(data, metrics = c("sens", "PPV"))
```

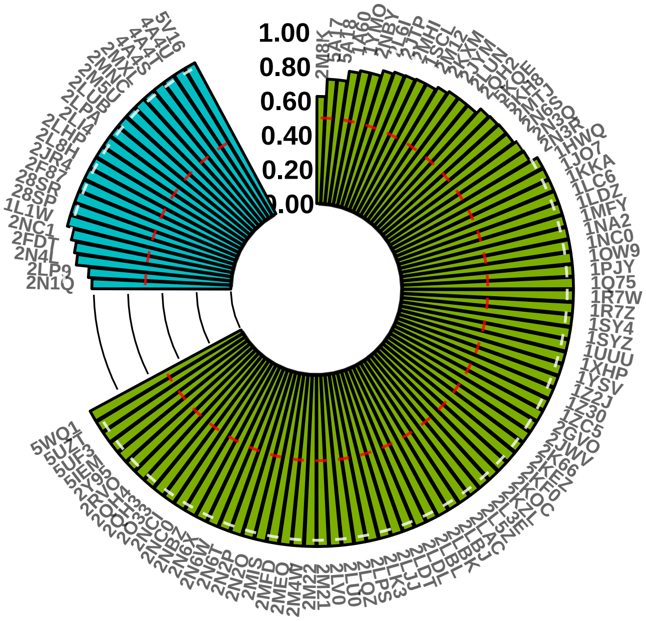
```
## metric:    sens min: 0.62 max: 1.00 mean: 0.97 median: 1.00
## metric:     PPV min: 0.71 max: 1.00 mean: 0.95 median: 1.00
```

```
plot_color_bar(data, "sens")
```

```
## Source: local data frame [2 x 5]
## Groups: <by row>
##
## # A tibble: 2 x 5
##   group  start   end title  mean
##   <fct>  <dbl> <dbl> <dbl> <dbl>
## 1 both       1    86  43.5 0.964
## 2 proton    97   118 108.  0.975
```

```
## Warning: Removed 20 rows containing missing values (position_stack).

## Warning: Removed 20 rows containing missing values (position_stack).
## Warning: Removed 20 rows containing missing values (geom_text).
```



```r
plot_color_bar(data, "PPV")
```

```
## Source: local data frame [2 x 5]
## Groups: <by row>
##
## # A tibble: 2 x 5
##   group  start   end title  mean
##   <fct>  <dbl> <dbl> <dbl> <dbl>
## 1 both       1    86  43.5 0.941
## 2 proton    97   118 108.  0.990
```

```
## Warning: Removed 20 rows containing missing values (position_stack).
```

```
## Warning: Removed 20 rows containing missing values (position_stack).
```

```
## Warning: Removed 20 rows containing missing values (geom_text).
```

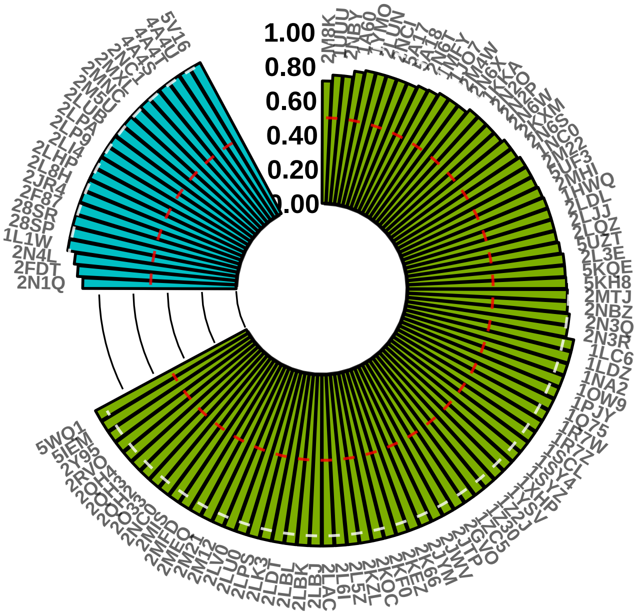


Table 1-1: base-pairing status prediction accuracy of different residue types

```
rnas <- get(load("info/rnas.RData"))
pred <- load_base_pairing_predictions(rnas, pred_cols = c("resid","bp_pred"), all_rnas = TRUE, method =
cat("Type\tTPR\tTNR\tInstances\n")
cat("G\t",residue_wise_cs2bps_metrics("G", pred, metric = "TPR"),"\t",residue_wise_cs2bps_metrics("G", p

cat("C\t",residue_wise_cs2bps_metrics("C", pred, metric = "TPR"),"\t",residue_wise_cs2bps_metrics("C", p

cat("A\t",residue_wise_cs2bps_metrics("A", pred, metric = "TPR"),"\t",residue_wise_cs2bps_metrics("A", p

cat("U\t",residue_wise_cs2bps_metrics("U", pred, metric = "TPR"),"\t",residue_wise_cs2bps_metrics("U", p
```

```
## Type TPR TNR Instances
## G     0.958    0.655    985
## C     0.965    0.561    859
## A     0.904    0.799    749
## U     0.916    0.641    759
```

Table 1-2: base-pairing status prediction accuracy of different base-pair types

```
pred <- load_base_pairing_predictions(rnas, pred_cols = c("resid","bp_pred"), all_rnas = TRUE, method =
cat("Type\tTPR\tInstances\n")
cat("GC\t",basepair_wise_cs2bps_metrics("AU", pred, metric = "TPR"),"\t",basepair_wise_cs2bps_metrics("A
cat("AU\t",basepair_wise_cs2bps_metrics("AU", pred, metric = "TPR"),"\t",basepair_wise_cs2bps_metrics("A
cat("GU\t",basepair_wise_cs2bps_metrics("AU", pred, metric = "TPR"),"\t",basepair_wise_cs2bps_metrics("A
```

```
## Type TPR Instances
## GC    0.88    772
## AU    0.88    772
## GU    0.88    772
```

Table 2: CS-Folding accuracy

select csfold structures based on: 1) cs2bps consistency; 2) folding energy; candidate structures are: Fold with and without cs2bps prediction; MaxExpect with and without cs2bps predictions; ProbKnot with and without cs2bps predictions.

copy final structure to 'ss_selected'

```
structure_selection_using_cs2bps_and_energy(rnas, from_path = "data/ss_with_cs/", to_path = "data/ss_sel
```

CS-Folding accuracy (with and without cs data)

```
Fold_with_cs <- load_scorer_accuracy("data/ss_scorer/FLpr1_prob_avg_scorer_summary.txt")
Fold_alone <- load_scorer_accuracy("data/ss_scorer/fold_scorer_summary.txt")
MaxExpect_with_cs <- load_scorer_accuracy("data/ss_scorer/MEpr1_prob_avg_scorer_summary.txt")
MaxExpect_alone <- load_scorer_accuracy("data/ss_scorer/maxexpect_scorer_summary.txt")
ProbKnot_with_cs <- load_scorer_accuracy("data/ss_scorer/PKpr1_prob_avg_scorer_summary.txt")
ProbKnot_alone <- load_scorer_accuracy("data/ss_scorer/probknot_scorer_summary.txt")
CSFold <- load_scorer_accuracy("data/ss_scorer/heuristic_scorer_summary.txt")

cat("Type\t\tTPR\t\t\tPPV\n")
cat("Fold\t",round(mean(Fold_alone$TPR),2),"/",round(mean(Fold_with_cs$TPR),2),"\t",round(mean(Fold_alon
cat("PK\t",round(mean(ProbKnot_alone$TPR),2),"/",round(mean(ProbKnot_with_cs$TPR),2),"\t",round(mean(Pro
cat("ME\t",round(mean(MaxExpect_alone$TPR),2),"/",round(mean(MaxExpect_with_cs$TPR),2),"\t",round(mean(
cat("CSFold\t",round(mean(CSFold$TPR),2),"\t\t",round(mean(CSFold$PPV),2),"\n")
```

```
## Type     TPR          PPV
## Fold  0.94 / 0.96     0.93 / 0.95
## PK    0.95 / 0.96     0.92 / 0.93
## ME    0.94 / 0.96     0.93 / 0.96
## CSFold    0.97        0.95
```

Table 3: CS-Folding TPR by base-pair type

```
data <- load_secondary_structure_predictions(rnas)
cat("Type\tTPR\tInstances\n")
cat("GC\t",basepair_wise_csfold_metrics("GC", data, metric = "TPR"),"\t",basepair_wise_csfold_metrics("
cat("AU\t",basepair_wise_csfold_metrics("AU", data, metric = "TPR"),"\t",basepair_wise_csfold_metrics("
cat("GU\t",basepair_wise_csfold_metrics("GU", data, metric = "TPR"),"\t",basepair_wise_csfold_metrics("
```

```
## Type TPR Instances
## GC    0.97    1378
## AU    0.98    772
## GU    0.9     184
```